NETPLAN a Network Planning Tool for Forecasting Trunks Dimensioning in a GSM Network

Technical Report No RR 96 026

Didier Samfat, Christian Bonnet
Institut Eurécom
2229, route des Crêtes,
BP 193
06904 Sophia-Antipolis,
FRANCE
Tel (+33) 93.00.26.26 Fax (+33) 93.00.26.26
{samfat, bonnet}@eurecom.fr
September 1996

Abstract - **In this report we present NETPLAN, a network planning tool for the GSM network. The work described herein is in fact an enhancement of the functionalities the WINES simulation project. Therefore, the goal of NETPLAN is to use WINES in order to allow network designers to perform an accurate forecast of future traffic and to re-dimension an existing network configuration for a given traffic load.**

Keywords: Simulation, protocols, GSM, trunk dimensioning, traffic forecast

TABLE OF CONTENTS

1 Introduction

The Global System for Mobile communications (GSM) is being more and more popular and has prompted the design of many new network applications and has stimulated operators to extend their networks. This cannot be achieved without a Network Planning Tool whose purpose is to provide network service planning, prediction of resources allocation and analysis of the network behaviour during overloading. As the creation of an overloading situation cannot be envisioned in a real GSM network, the integration of a WIreless NEtwork Simulator (WINES) in the design of NETPLAN will allow network designers to perform an accurate forecast of future traffic and to re-dimension an existing network configuration for a given traffic load.

However, NETPLAN is not a tool for cellular planning (like PLANET) but its goal is to serve as a help in the design of the Network Sub-System (NSS) process. Therefore, the main functions of NETPLAN are the following:

- Traffic forecast
- Trunk dimensioning
- Equipment dimensioning
- Optimization
- Simulation of a whole GSM network
- Common channel signalling dimensioning
- Graphical Network Display
- Reporting

2 The Network Simulator WINES

The GSM network simulator WINES has been developed with OPNET software [17]. The resulting platform model simulates the BSS, the NSS and the MS as described in section 2. All layer-3 messages exchanged by the different GSM entities have been implemented in accordance with the GSM recommendation. The global architecture is depicted in Figure 2.

All GSM entities of the NSS and the BSS are OPNET network nodes; each network node is composed of one or several processors[1] and of several emitter-receiver pairs for the network connection. Furthermore, each processor is associated to a finite state machine (FSM) whose main function is to manage the different GSM protocols (RIL-3, MAP, BSSMAP) [15], [16]. In order to reduce the simulation time and test the platform for the intrusion detection architecture, all communications are assumed to be reliable[2].

2.1 GSM Network Entities

A BTS has several attributes which help in defining the characteristics of the cell to be chosen upon a handover during the target cell selection procedure. Furthermore, a «beacon frequency»

1. *Depending on the complexity of the GSM component*
2. *The loss of messages is not yet simulated*

attribute may also be chosen for each BTS. The radio equipment of a BTS is connected to 5 links representing 4 dedicated channels and one common channel allowing simultaneous communications with the mobile stations.

A BSC manages a table of allocated channels where each entry in the table is a structure describing the availability, the connection identification with a BTS (or MSC) and the type of a channel.

A MSC has a connection table where each entry is also a structure describing the active connections and other informations relevant to the call control and handover processing. In the platform, the VLR is integrated to the MSC node as it is typically done in real GSM networks.

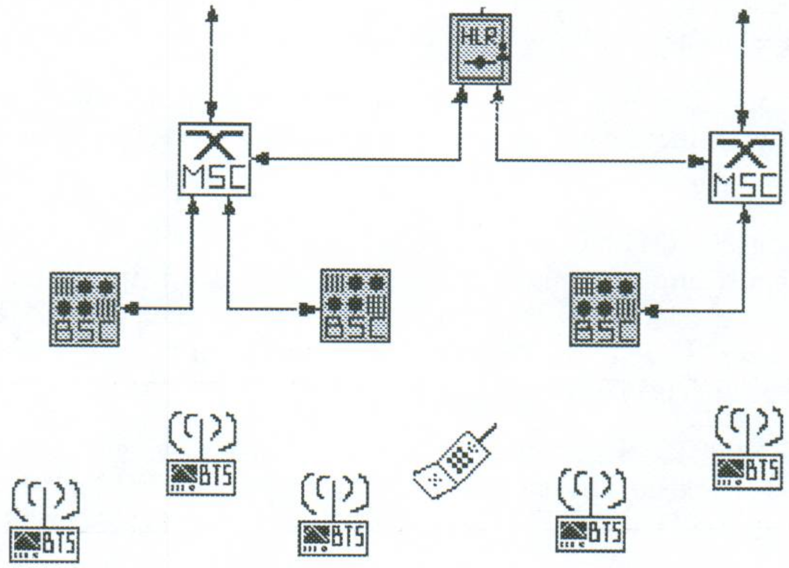

Figure 1 : A Portion of the basic version of WINES.

In addition to the GSM entities, two special nodes have been represented:

- The PSTN node (Public Switching Telephony Network) for the simulation of mobile originating and terminating calls.
- The SS7 node (Signalling System number 7) has been implemented in order to connect different public networks.

The SS7 protocols has been simplified[1] in order to reduce the simulation time. The main function of the SS7 node is to route signalling messages towards the NSS entities and the PSTN.

The simulator is able to perform common procedures specific to cellular mobile networks. We briefly describe these procedures which require the participation of the different GSM entities.

1. *Only a light version of the Signalling Control Connection Part protocol and the Transaction Capabilities Application Part protocol have been implemented*

2.2 Wireless Network Procedures

Location updating (locup). This procedure is triggered by the MS each time its current location area changes. The corresponding information is sent by the BTSs on their control channel. The goal of this procedure is two folds: first, it allows the HLR to know which MSC is currently in charge of the MS, second, it registers the MS in the VLR of the current MSC. The location updating request emitted by the MS is processed by the new MSC:

- If the old location area was already under its control, an ***inter-BSC-locup*** procedure is performed and the MSC has only to update the VLR. However
- If the old and the new location areas are under the control of separate MSCs, an ***inter-MSC-locup*** is performed. In this case, the old MSC informs the HLR of the MS registration in the new VLR, as well as its entry cancellation in the old VLR.

Mobile Originating call (MO). The mobile user can make a call towards the PSTN; the frequency and duration of these calls are parametrizable. The MSC relays the call request to the PSTN and establishes the connection[1].

Mobile Terminating call (MT). Calls can also be initiated by the PSTN which asks the Gateway MSC for routing information. Therefore, the GMSC asks the HLR for the roaming number provided by the current VLR and routes the call toward the current MSC. The MS is then paged in the location area where it was previously registered. Upon receipt of the reply from the MS, the MSC establishes the connection between the PSTN and the MS.

Handover. During a call, the BSC continuously analyses the measurement results sent by the BTS in order to decide on a potential handover. The following procedures are performed depending on the type of handover:

- ***internal-handover***: the MS moves from a current cell to a target cell both managed by the same BSC
- ***inter-BSC-handover:*** the MS moves to a target cell managed by another BSC in the same MSC area
- ***inter-MSC-handover:*** the MS moves from the anchor MSC[2] area toward a target cell in another MSC area (relay MSC area)
- ***subsequent-handover***: the MS moves from the relay MSC area to another relay MSC area, or is back to the anchor MSC area

2.3 Probes and Spy functions

In order to design an efficient planning tool, we have added to WINES a set of new functions for the analysis of the generated traffic. Therefore, three types of probes (or spy-functions) have been implemented at different levels of the GSM network: BTS, BSC and MSC nodes.

---

1. *Speech data traffic is not yet simulated*
2. *MSC managing the MS at the beginning of the call*

These functions can help operators to assess the following network planning criteria: trunk dimensioning equipment dimensioning or common channel signalling dimensioning is described below in Section 4.

Moreover, these probes update local statistics for each entity of the network; for instance, we are able to collect the following relevant information:

- Instantaneous traffic load in terms of dedicated channels such as Traffic Channel at Full rate (TCH/F) and Standalone Dedicated Control Channel (SDCCH)
- Number of handovers performed (or failed) by a specific entity in the network
- Number of location updates performed (or failed) at different levels of the network

Therefore, we can measure the number of traffic channels and SDCCHs allocated to each BSC and MSC composing the simulated GSM network. In other words, we are able to represent the load variation on each link (in terms of number of dedicated channel manage by each entity) as a function of time.

3 Mobile Station Generators (MS-GEN)

The complexity of the mobile station model of WINES forbids its duplication for the simulation of a wide network traffic because such approach requires a simulation time of several days. Network planning application which are essentially based on the analysis of the NSS traffic do not require the simulation of the radio part.

Type of User	DOMESTIC	CORPORATE	LOCAL BUSINESS	ROAMER
Daily Usage [a]	10-15	30-40	15-20	30 +
Calls per Week	12-16	20-25	12-15	25 +
National Call	95%	75%	99%	95%
International Call	5%	25%	1%	5%
Average Duration[b]	< 5	10	< 10	10 +
Call Time	Off peak	Peak Time	business Hours	Any
Destination Call	Mostly local	Nation Wide	Local	International
Origine of Call	Home Cell	Any MSC/BSC	Same MSC	All
Type of Call	MTL[c]	MTL and **MTM**[d]	MTL	MTL and MTM

Table 1 : Classification of GSM subscribers

a. *Value in minutes*
b. *Value in minutes*
c. *Mobile to Land*
d. *Mobile to Mobile*

Therefore, WINES has been modified in order to reduce computation time. In doing so, we made the design of a Mobile Station Generator (MS-GEN) which does not use the radio part. Only the protocols involving the BSC and the NSS entities which manage the telephony activity and the migration of mobile stations are simulated.

3.1 The Network Model

The goal of a MS-GEN is to simulate a population of mobile stations in order to generate realistic signalling and user data traffic within WINES. The characteristics of the generated traffic must be as closed as possible to the real traffic for a given GSM topology. Therefore, each MS-GEN is configured in order to activate a particular category of real GSM subscribers[1] as described in Table 1.

Each MS-GEN is able to simulate a group of 1000 mobile stations with the same behaviour. The frequency and the duration of MO calls are parametrizable and the peak hours (busy period during the day) can also be defined. For each user, a scenario of behaviour is preestablished at the beginning of the simulation depending on the following statistical distributions:

- Duration of calls follows an exponential distribution
- Arrival of calls follows a poisson distribution

These assumptions have been shown to be reasonably accurate for mobile telephone networks. In addition to the telephony activity, the parameters corresponding to the mobility of users can be defined:

- Number of handovers per day
- Number of location updates
- Number of location areas traversed

A simulation run can represent the behaviour of the network during one day. Therefore, a MS_GEN is able to simulate a population of MSs during work days or week-ends. The different MS-GENs can be run in parallel or in sequence:

- The parallel mode allows the network designer to obtain at the end of the simulation, relevant information about the behaviour of the GSM network during the day.
- The sequential mode allows the analysis of the variation of the network load in an incremental way. For instance, it is possible for the network designer to measure the impact of an additional ten thousand MSs on the network for a given GSM topology.

The MS-GEN has been designed in order to be independent of the underlying simulated network and can be used for the generation of traffic in other wireless network simulator. However, using a MS-GEN requires that the normal BTSs be replaced by a derived entity called RELAY which translates mobile units activity[2] in terms of signalling messages. However, like a normal BTS a RELAY is connected to a BSC via the A-bis interface. Moreover, each RELAY manages a connection table and routes messages between the BSC and the MS-GEN in charge of the current mobile station.

---

1. *This classification of GSM subscribers was obtained from a french operator*
2. *A MS-GEN generates locup, handover an MO/MT calls events which must be transformed into the corresponding signalling protocol that the target simulator can understand*

3.2 The Node Model

Each generator has a one emitter/receiver pair which are used to receive/transmit packets from/to the relays. Moreover, the transmission mode used is the «hyperspace» mode which does not require a propagation model and thus minimizes the simulation duration. The node model attributes are the following:

- «*MS number*»: this is the total number of MS that the MS-GEN is going to simulate
- «*Classmark*»: this attribute represents the maximum transmission power and the services the network can support. For instance, if the classmark is 2 the maximum transmission power of the MS will be 8 watts.
- «*Home PLMN*»: this is the identification number of the PLMN, ranging from 1 to n (integer). Note that this attribute must be set accordingly as all ISDN address will be computed using this value.
- «*First User Id*»: this is the identification number of the first MS, ranging from 1 to n (integer). Note that this attribute must be set accordingly because if 2 MS-GENs have the same value for this attribute then the simulator will simulate clones of MS as well as normal MS. This will lead to lots of errors during the simulation
- «*peak starting hour*»: this attribute represents the beginning of the peak hours. Putting this attribute to 10.0 means that the peak hours will start at 10h00 AM.
- «*peak ending hour*»: this attribute represents the end of the peak hours. Putting this attribute to 18.0 means that the peak hours will start at 18h00 PM.

At the node level, a MS-GEN is composed of a ***trajectory***, ***behaviour*** and ***main*** processor.

The ***trajectory*** processor is responsible for managing the migration of the simulated MS. The trajectory of each MS is computed at the beginning of the simulation. In fact, each trajectory is composed of a list of RELAY that the trajectory processor will manage. Therefore, each time a MS will roam into a new location area, this processor will signal the ***main*** processor of this new action. The following parameters of this processor can be initialised at simulation time:

- «*min number cells*»: this attribute represents the minimum number of cells that each MS is going to cross (number of location update).
- «*max number cells*»: this attribute represents the maximum number of cells that each MS is going to cross (number of location update).
- «*handover number*»: this attribute represents the total number of handovers that the MS is going to perform during the simulation. If this attribute is not null, then each time the MS will start a MO call then the ***force_handover()*** function (defined in the function block) will be triggered. Note that this number should always be smaller than the number of calls.

For each MS, the number of locup to be triggered is chosen randomly according to the OPNET statistical uniform distribution.

The ***behaviour*** processor is responsible for managing the telephony activity of each MS. In other word, this processor is responsible for switching on the MS and triggering a MO calls. The following parameters can be set at simulation time:

- «*min duration*»: this attribute represents the minimum duration of each call in seconds
- «*max duration*»: this attribute represents the maximum duration of each call in seconds
- «*min arrival*»: this attribute represents the minimum number of call to be triggered during the simulation
- «*max arrival*»: this attribute represents the maximum number of call to be triggered during the simulation
- «*peak hours flag*»: this attribute is a boolean variable and represent whether the MO calls should be triggered during peak hours or not. Putting this flag to «enable» means that all calls will be performed during peak hours
- «*switch on time*»: this attribute represents the moment during the day (i.e each simulation) when the MS should be activated (switched on).

In addition, different type of MO calls can be simulated and are defined by the following boolean variables:

- «*called party busy*»: this attributes represent the MS trying to call a PSTN subscriber who has already engaged a conversation with someone else. Putting this flag to «enable» means that all MO calls will never be fully completed.
- «*called party missing*»: this attributes represent the MS trying to call a PSTN subscriber who is not reachable (not at home) and therefore cannot hang up the phone. Putting this flag to «enable» means that all MO calls will be aborted after a ringing period.
- «*called party non exist*»: this attributes represent the MS trying to call a PSTN subscriber who does not exist (wrong number). Putting this flag to «enable» means that all MO calls will be aborted by the PSTN node.

Note that the duration of calls follows an exponential distribution and the arrival of calls follows a poisson distribution as mentioned in Section 3.1.

The ***main*** processor is responsible for managing the inter-operability of the ***trajectory*** and the ***behaviour*** processor. Moreover, this processor manages a list of MS (table) where each entry contains a pointer to a special process models called «***Gsm_ms***». The Gsm_ms Finite State Machine (FSM) is in fact a light version of the MS models defined in the first version of WINES. Therefore, if a MS-GEN is simulating 10 MS the list will contain 10 pointers; in other words, the MS-GEN will manage 10 FSMs.

3.3 The MS_gen_trj Process Model

1. «***init***» state

   In this state we program the mobility of the user. Each MS will trigger a locup upon crossing a relay (i.e a cell). The relay list and the MS list are both initialized in the *init* state. The relay list is a table of structures of **relay_description_type** (see generator.h) containing for each relay of the network its object id, its beacon frequency, and the list of all its neighbours. The MS list is a table of structures of **MS_description_type** (see generator.h) containing the following information:

- IMSI,
- current Status,
- channel reference and relay,
- the description of its trajectory
- various flags.

The trajectory is defined as an ordered set of relays. The initial position of the MS is chosen randomly among the relays in the MSC area determined by the IMSI and the number of MSC. In doing so, the MS can be activated through out the whole PLMN.

Note that the HLR database (referenced as "hlr.input data file" attribute of the HLR node) must already contained an entry for all MS simulated by the MS-GEN. Moreover, this file must also indicate where the MS have performed their last location update (MSC area).

2. «***MS on***» state

   In the «init» state the time when the MS will switch on is computed an schedule. Upon receiving the **SWITCH_ON_CODE** interruption, the MS-GEN will activate all MS by putting the flag MS_list[i].Status to **ON.**

3. «***MS dedicated***» state

   The process enters the *MS dedicated* state upon receiving a remote interruption (**MS_DEDICATED_CODE**) from one of the children of the *main* process. The index of this MS in the list is read in the associated ICI. Then, the corresponding MS sends this signal once a traffic channel (with a reference contained in the ICI), has been assigned. This channel reference is afterward stored in the **MS_list** table.

   The process resets both the MS current channel reference and Connected flag, in the *stop* state upon receiving a **STOP_MEASUREMENTS_REPORTS** interruption (this means that the current channel is no used any more).

4. «***MS connected***» state

   Once a connection has been established for a MO or MT call, the MS sends a **MS_CONNECTED_CODE** interruption. The FSM enters the *MS connected* state and sets the **Connected_flag** of the **MS_list** table to TRUE for the MS designated in the ICI. Then, if the «*handover number*» attribute has been set (i.e, > 0) the FSM will call the force_handover() procedure in order to trigger an handover.

5. «***Compute & send***» state

   In this state we force the measurements results in order to trigger handovers or locup. In other words, as the radio interface is not simulated any more, no quality of reception measurement is performed. Therefore, according to the scenario set at simulation time (i.e, number of cells crossed) a set of location update procedures will be triggered. Moreover, if a MO call has taken place before entering this state then a handover will be triggered.

In general, a new position is computed (see new_pos() in the function block) for each active MS. The computation consists in choosing randomly and with the same probability, one of the current relay's neighbours as the new relay. The field LA_counter of the MS_list table will be incremented if the current location area has effectively changed.

If the MS is in the connected phase, a measurement report will then be sent to the BSC. If the MS current relay has not changed, the report will be: good measures for the current cell and very bad measures for the other cells. Otherwise if the current relay has changed, the report will be made so as to immediately provoke a handover involving the new position (with very bad measures for the current cell and very good measures only for the first neighbour cell). This is possible only if the BSC parameters Hreqave, N5, P5, N6, P6, N8 and P8 are all set to 1, thus allowing the BSC to store only one averaged value of the measurements, and to take the handover decision after the reception of only one measurement report.

After the measurement report has been set up, an interruption is sent to the *main* process (with code HANDOVER_EXPECTED_CODE), associated to an ICI containing the new relay id and its beacon frequency. Note that if no location updating or IMSI Attach ever occurred before the FSM will send a IMSI_ATTACH_CODE interruption to the main process in order to trigger a locup.

If the current relay has changed but not the location area, a CELL_CHANGE_CODE interruption will also be sent to the main FSM. Otherwise, if the location area has changed LA_CHANGE_CODE interruption will be sent to the main FSM.

All the remote interruptions destined to the *main* process are sent in the force mode i.e, op_intrpt_force_remote(), in order to have the state variables (such as current relay id, Busy_flag, etc...) immediately modified. This avoids interferences with any call that would start at the same moment.

6. «***End***» state

   If the duration variable set at simulation time is not big enough to handle the desired scenario, a message is displayed for each MS whose trajectory is not finished yet.

3.4 The MS_gen_bhv Process Model

The parameters of the MS behaviour depends on a table like Table 1. For each category of users we have a range of values for call duration and number of calls per day while taking into account whether the user makes call during peak hours or not.

1. «***init***» state

   In this state the different distributions are loaded and the following variables set:

   - Call type: this variable can have the following values: PSTN_USER_NUMBER (a normal successful call will be performed), PSTN_USER_UNREACHABLE_NUMBER (callee not at home to hang up phone), PSTN_USER_BUSY_NUMBER (callee has already engaged a conversation) or PSTN_USER_UNKNOWN_NUMBER (callee is an unknown PSTN subscriber).

- Number of calls per day: this variable is set according to the «min arrival» and «max arrival» attributes.
- Average duration of each call: this variable is set according to the «min call duration» and «max call duration» attributes. The value obtained is converted into OPNET simulation time.
- Starting and ending OPNET time for peak hours: time when to start and end peak hours. The value obtained is converted into OPNET simulation time.

Note that if the simulation duration is not long enough to performed all calls with the specified duration, a warning message will appear. In most cases, NETPLAN can recover the error but I won't bet my life on it.

2. «*Switch on*» state

   Upon receiving a self **SWITCH_ON_CODE** interruption, the FSM enters the *switch on* state. Then, the first MO call will be scheduled according to the current arrival distribution and whether the calls should be performed during peak hours or not. Note that the FSM will enter this state only once because the next MO calls will be scheduled in the *call start* state.

3. «*Call start*» state

   Upon receiving a self **MO_CALL_CODE** interruption, the FSM enters the *call start* state. The process re-uses the same ICI (containing the MS index, the service type and the called number) in order to send a remote **MO_CALL_CODE** interruption to the main process.Then, it schedules the end of the call (self **CALL_END_CODE** interruption) with a delay computed from the current call duration distribution. The corresponding event handle is stored in the **call_end_evh** table.

   The call might be rejected by the main process, which send back a **CALL_REJECT_CODE** interruption. In this case the process enters the *call reject* state, cancels the end of the call, and schedules another MO call 3 seconds later.

4. «*Call end*» state

   When the **CALL_END_CODE** interruption occurs, the process enters the *call end* state and transmits the same interruption with the same ICI to the *main* process. Then, it schedules the beginning of the next MO call, with a delay computed from the current call period distribution (i.e delay between 2 MO calls).

   The end of the call might be rejected by the *main* process if the MS is performing a handover at that moment. In that case, a **CALL_END_REJECT_CODE** interruption is returned to the *MS_gen_bhv process* which enters the *end reject* state in order to schedule once again the end of the call 1 second later.

5. «*Peak hours*» state

   Upon receiving a self **START_PEAK_HOUR_CODE**, the process enters the *start peak hour* state and sets the current call period distribution accordingly. When the **END_PEAK_HOUR_CODE** occurs, the process enters the *end peak hour* state and sets back the current call period distribution to the off peak period distribution.

3.5 The MS_gen_main (Father) Process Model

The MS_gen_main FSM is the core of the MS-GEN. It is responsible for coordinating MS_gen_bhv, MS_gen_trj and the child process Gsm_ms.

1. «*init*» state

   Each MS is a single child process which is the Gsm_ms FSM. The MS_gen_main FSM manages a list of child process which is a table of **child_process** structure (see generator.h). Each entry of this table contains the following information:

   - the process handle,
   - the MS status (on or off),
   - the current channel reference and relay id used,
   - additional data about the MS current actions.

   Depending on the number of MS that the generator is going to activate memory is allocated for the creation of a child process table. Then, we schedule the moment when the different Gsm_ms should be activated.

   Note that the FSM of each child process is neither created nor invoked in this state. This will be done after the father process receives the SWITCH_ON_CODE interruption.

2. «*MS enable*» state

   The FSM activates this state upon receiving the self SWITCH_ON_CODE interruption. Therefore, the MS_GEN creates a table (or list) of MSs while positioning activity and mobility flags. Then, for each MS the MS_GEN creates a child process which FSM is defined in «Gsm_ms». At the end of this state, the «main» FSM allows the «behaviour» and «trajectory» FSM to start locup and MO calls.

   Note that the child process are invoked just after their creation in order to save their table index in the "parent-to-child" memory. This memory is in fact a table (**parent_child_memory[]**) which is used by the father process to communicate with the invoked child (there is one entry for each child in the table).

3. «*forward*» state

   All packets arriving to the generator have only the RIL3 protocol format. The associated ICI contains the channel reference in addition to the source relay id which are enough to find the corresponding MS in the **child_process[]** table.

   Then, the packets are directly forwarded to the corresponding MS as followed:

   - the child process is invoked,
   - the interruption code takes the value of the packets's **MessageType** field
   - the address of the packet is stored in the "parent-to-child" memory

   Depending on the nature of the packets the following actions akin to the RIL3_RR protocol can take place:

   - ***Immediate Assignment***: this is the typical case where the MS index has not been saved in the table yet because it is the first time that the channel reference is used.

Therefore, the search consist in looking for the right relay id and the Waiting_channel_flag set to TRUE. If the result is successful then the current channel reference is updated and the packet is forwarded to it.

- ***Paging Request Type 1***: the message is forwarded to every MS whose current relay is the one which sent the paging request. In this case, knowing the MS identity is useless to the father process.
- ***Handover Command***: the current relay id and channel reference of the concerned MS are updated according to CellDescription and ChannelDescription fields of the received packet. This is done after checking that this cell description (the beacon frequency) corresponds to the one previously sent by the *MS_gen_trj process*. The locup_needed_flag of the child_process table is set to TRUE.
- ***Location Updating Accept***: the current relay id of the concerned MS is updated.
- ***Assignment Command***: the current channel reference of the concerned MS is updated.
- ***Channel Release***: the Busy_flag of the concerned MS is set to FALSE. If the Locup_needed_flag is set to TRUE, the process triggers a location updating by sending a self LA_CHANGE_CODE.

4. «***from MS***» state

   The MS common actions (location updating, calls, handover) are triggered by the *trajectory* or the *MS_gen_bhv process* via a remote interruption sent to the father process. The ICI associated with this interruption contains the necessary informations. The father process then invokes the concerned child process, using the same interruption code and the same ICI.

5. «***locup***» state

   The FSM activates the «locup» state upon receiving a LA_CHANGE_CODE or IMSI_ATTACH_CODE from the *MS_gen_trj process*. Then, the father process reads the MS index and the new relay id in the ICI and invokes the child process with the same interruption.

   The FSM starts the access procedure by sending a RSM Channel Required message to the BSC (the access on RACH is not simulated). This is the only case the RSM protocol is used by the generator because other RSM messages are handled by the relay.

   Finally, the FSM enable the Busy_flag (set to TRUE) and indicates that the MS is waiting for an assignment from the new relay. In other words, the new relay id is set in the Relay field and the Waiting_channel_flag is set to TRUE.

6. «***Call start***» state

   The FSM activates the «Call start» state upon receiving a MO_CALL_CODE interruption from the *MS_gen_bhv process*. If the MS is not busy, the following actions take place:

   - the child process is invoked
   - the channel activation is required from the BSC,
   - the MS is marked as busy and waiting for an assignment

If the MS is busy the call is rejected and a CALL_REJECT_CODE interruption is sent back to the *MS_gen_bhv process*.

7. «***Call end***» state

The FSM activates the «Call end» state upon receiving a CALL_END_CODE interruption from the *MS_gen_bhv process*. The child process concerned is invoked if the corresponding MS is not currently performing a handover. Otherwise the end of the call is rejected and a CALL_END_REJECT_CODE interruption is sent back to the *MS_gen_bhv process*.

8. «***IMSI detach***» state

The FSM activates the «IMSI detach» state upon receiving an IMSI_DETACH_CODE interruption from the *MS_gen_trj process*. The Willing_to_detach_flag is set to TRUE indicating that the concerned MS is willing to switch off. If the MS is busy at that moment, nothing will be done until the Busy_flag is set to FALSE again (i.e, upon receiving a RIL3_RR Channel Release).

However, if the MS is busy and waiting for a channel assignment the child process is invoked with the same IMSI_DETACH_CODE interruption. Then, the child process will send a RIL3_MM Imsi Detach message after receiving the assignment and the «switch off» interruption.

Otherwise, if the MS is not busy, the father process will invoke the child process by requesting a channel allocation. Both Willing_to_detach_flag and Busy_flag are set to TRUE.

9. «***Cell change***» state

The FSM activates the «Cell change» state upon receiving a CELL_CHANGE_CODE interruption from the *MS_gen_trj process*. If the current cell has changed but not the current location area, the current relay id is updated and the concerned child processed is invoked with the same interruption code providing that the MS is in the idle mode.

10. «***Handover***» state

The handover decision is taken by the BSC, but it is the *MS_gen_trj process* that triggers the corresponding procedure by sending the appropriate measurement reports. This is why the *MS_gen_trj process* can inform that a handover is going to occur before any received message from the network.

The FSM activates the «Cell change» state upon receiving a HANDOVER_EXPECTED_CODE interruption from the *MS_gen_trj process*. The data contained in the ICI are both stored by the father process and forwarded to the child process. The Handover_expected_flag from the child_process table is therefore set to TRUE.

11. Other interruptions

Other interruptions may be received directly from one of the children processes (see state *from MS*). After receiving a the RIL3_RR Paging Request message, an ASK_FOR_CHANNEL_CODE interruption is sent by the MS in order to ask the father process to start the access procedure by sending a RSM Channel Required message

Another interruption with code SWITCH_OFF_CODE is sent by a MS after having sent a RIL3_MM Imsi Detach message to the network. The father process sets the status of the concerned MS to OFF.

Finally, the MS sends a **HANDOVER_PERFORMED_CODE** interruption after the end of a handover in order to have the **Handover_expected_flag** of the **child_process** table reset to FALSE.

3.6 The Gsm_ms (child) Process Model

Each MS is the child process of the MS_gen_main FSM referenced as the **Gsm_ms** model. The most important state of the process is the central *idle updated* state which corresponds to the idle state of the MS. On the left side of the «idle updated» state lie the states concerning calls and handover whereas on the right sides we have the states concerning location updating.

Note that this FSM has not been modified since its creation in october 1994 (original description in [4]).

1. «***init***» state

   Upon invocation for the very first time the child process, the Gsm_ms FSM executes the *init* state. The FSM performs the following actions:

   - it reads its own index in the "argument memory", used as an argument of the Opnet primitive op_pro_invoke()
   - it initializes some variables
   - it enters the *idle updated* state.

2. «***locup***» state

   The procedure starts with the arrival of the **LA_CHANGE_CODE** or **IMSI_ATTACH_CODE** interruption. The old relay id and new relay id contained in the ICI are stored and the process enters the *locup* state. The father process as already asked a channel so, the MS needs only to wait for the assignment.

   Upon receiving a RIL3_RR Immediate Assignment message, the process reads the reference of the assigned channel and sends back a RIL3_MM Location Updating Request. It then enters the *waiting update* state.

   If the network answer is positive (i.e, RIL3_MM Location Updating Accept) the process updates the current relay id and enters the *RR release 1* state. Otherwise, if the location updating failed (RIL3_MM Location Updating Reject) it enters the *RR release 2* state. In both cases the release occurs upon receiving a RIL3_RR Channel Release message; then process enters either the *idle updated* or the *idle not updated* state. In the last case the MS is not allowed to perform any call, but it may start a location updating procedure again.

3. «***MO call***» state

   The MO call establishment takes place in the upper part of the diagram, from the *idle updated* state to the *on line* state.

   Upon receiving a **MO_CALL_CODE** interruption, the process reads the data contained in the ICI (called number & service type, necessary for the RIL3_CC Setup message) and transits from the *idle updated* state to the *MO call* state.

   Like in the location updating case, the child process awaits an assignment. After receiving this assignment (through RIL3_RR Immediate assignment) it sends back the RIL3_MM CM

Service Request message, before entering the *dedicated* state (the MS is in dedicated mode).

Then, the MS sends the RIL3_CC Setup message, and on reception of the RIL3_CC Call Proceeding answer it enters the *proceed* state.

The subsequent assignment procedure begins with the reception of the RIL3_RR Assignment Command message. The child process updates the current channel reference and sends back a RIL3_RR Assignment Complete message. The child process also informs the MS_gen_trj process of the current channel reference via a remote MS_DEDICATED_CODE interruption and the associated ICI.

It enters the *new assignment* state, then the *alert* state, upon receiving a RIL3_CC Alerting message. After sending back a RIL3_CC Connect Acknowledge message, the child process enters eventually the *on line* state on receipt of the RIL3_CC Connect message. Then, it informs the MS_gen_trj process of the connected state of the call, by sending a remote MS_CONNECTED_CODE interruption.

4. «***MT call***» state

The MT call establishment is quite similar to the MO call procedure. However, it takes place in the lower part of the diagram, from the *idle updated* state to the *on line* state.

Upon receiving a RIL3_RR Paging Request message containing its own IMSI, the process leaves the *idle updated* state and enters the *MT call* state. Then, it sends a ASK_FOR_CHANNEL_CODE interruption to the father process in order to have a channel assigned by the network.

Upon receiving the RIL3_RR Immediate Assignment message, and if no IMSI Detach has been required, it sends back a RIL3_RR Paging Response message, to the BSC, and enters the *dedicated* state. Then, the process transits to the *confirmed* state on receipt of the RIL3_CC Setup message.

The subsequent assignment procedure begins with the arrival of the RIL3_RR Assignment Command message and the following actions are performed:

- the current channel reference is updated
- a RIL3_RR Assignment Complete message is returned to the network
- a MS_DEDICATED_CODE interruption is sent to the *MS_gen_trj process* before entering the *new assignment* state.

Because the «MT call» state is a forced state, the process has to perform the following actions:

- it sends a RIL3_CC Alerting message
- it send a RIL3_CC Connect message,
- it enters directly the *alert* state waiting for the connection acknowledgement.

Upon receiving this acknowledgment, the FSM enters the *on line* state and informs the *MS_gen_trj process* of the connected state of the call.

5. «***Handover***» state

When the *MS_gen_trj process* has decided to trigger a handover, the FSM sends a

HANDOVER_EXPECTED_CODE interruption to the main process which forwards it to the concerned child process. The child process stores the new relay id contained in the ICI and remains in the *on line* state. The purpose of this step is to let both the father and the child process know about the relay id of the target cell. This is done because the target cell is not fully described in the RIL3_RR Handover Command message, but described in terms of beacon frequency.

A RIL3_RR Handover Command message must arrive. The current relay id and channel reference are updated. A RIL3_RR Handover Complete is then sent back to the BSC and two remote interruptions are sent:

- one to the MS_gen_trj process (code MS_DEDICATED_CODE) to inform it about the new channel reference
- the other one to the father process to confirm handover.

The process immediately returns to the *on line* state afterwards.

6. «***Call release***» state

The call release takes place in the middle part of the diagram, from the *on line* state back to the *idle updated* state.

If the mobile has initiate the release, a CALL_END_CODE interruption has first been received from the father process. The child process performs the following actions

- it enters the *MO release* state
- it sends a RIL3_CC Disconnect message to the network
- it sends a STOP_MEASUREMENTS_REPORT_CODE interruption to the *MS_gen_trj process*.

Then, upon receiving a RIL3_CC Release message, it will send back a RIL3_CC Release Complete and enter the *RR release* state.

Otherwise, if the release is initiated by the network, the process enters the *MT release* state upon receiving a RIL3_CC Disconnect message. In that case, the FSM performs the following actions:

- it sends back a RIL3_CC Release message
- it enters the *RR release* state upon receiving a RIL3_CC Release Complete acknowledge.

The RR release takes place when a RIL3_RR Channel Release message arrives; the current channel reference is reset to a negative value.

7. «***Imsi Detach***» state

If a IMSI_DETACH_CODE interruption occurs while the MS is in the *idle updated* state, the process enters the *IMSI detach* state, sets the imsi_detached_required flag to TRUE, and returns to the *idle updated* state. Otherwise, if the interruption occurs while the MS is in the *MO call* or *MT call* or *locup* state, the imsi_detached_required flag is set to TRUE and the process remains in this state.

In both cases when the RIL3_RR Immediate Assignment message is received, the process performs the following actions:

- it enters the *idle updated* state,
- it sends the RIL3_MM Imsi Detach as initial message,
- it sends a **SWITCH_OFF_CODE** interruption to the father process.

The **IMSI_DETACH_CODE** interruption should never occur at any other moment since it is filtered by the father process.

4 Integration of WINES into NETPLAN

The GSM platform has been integrated into NETPLAN network planning tool. Because configuration, accounting and fault management functions are present in various management environments we have provided NETPLAN performance and network planning functions which are more difficult to implement as they require data collection and thus create computational overhead.

4.1 Overview

Simulators based on WINES approach are of interests to operators because they can address two different needs at the same time:

- Replay of realistic scenarios based on accurate observations made on a real network in order to find the best configuration
- Experiment of hypothetical growing traffic in order to predict the resources to be added to the existing network

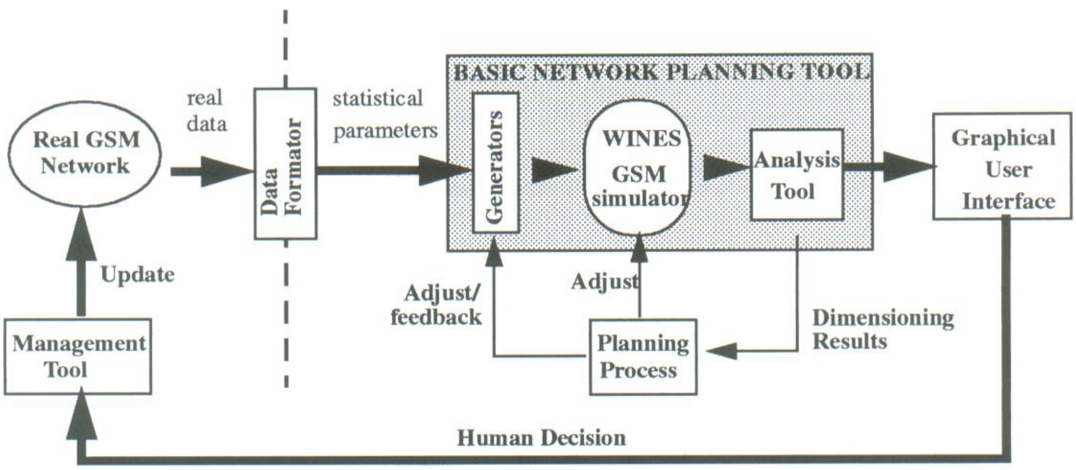

Figure 2 : Overview of NETPLAN

Therefore, WINES has been integrated into a basic network planning tool in order to allow GSM operators to design the future extension of their BSS and NSS. The basic network planning tool is depicted in Figure 2 and is composed of several[1] Generators, WINES (for the simulation of the BSS and NSS of GSM) and of an analysis tool which presents the output results in the form of histograms or graphs.

1. *depending on the number of mobile stations to be simulated*

Data are collected from a real GSM network and must be formatted in order to initiate the different Generator variables as well as the parameters of statistical laws as mentioned in Section 3. Then, each Generator activates a set of mobile stations which generate signalling messages in both BSS and NSS.

4.2 Traffic Forecast

NETPLAN is able to perform an accurate prediction of future traffic on the basis of input parameters of the Generator as described in Section 3.2:

- The traffic per subscriber in busy hour can be expressed with parameters: calls duration, calls arrival, and peak hours flag.
- Total number of MS per region (the region can be a PLMN) which is the sum of all number of MS of each MS-GEN can be expressed with parameter «MS number»
- Traffic generated by PSTN subscribers can be expressed with parameters: PSTN call duration, PSTN call arrival
- Percentage of originating and terminating MS calls can be set according to MS number parameter of each MS-GEN and the PSTN user number which is the PSTN_PROCESSING_CAPACITY defined in Gsm.h
- Percentage of local an regional calls (both originating and terminating): a specific generator can be dedicated to simulate these calls according to the GSM topology modelled
- Percentage of PLMN to PSTN calls (see above)
- Percentage of PSTN to PLMN calls (see above)

Note that the following parameters have not been taken into account in the current version of the platform:

- Percentage of PLMN to PLMN calls
- Percentage of international calls (both originating and terminating)
- Percentage of VMS calls per subscriber, PSTN, PLMN and VMS originating calls
- Percentage of ACD calls per subscriber (both originating and terminating)
- Percentage of data and fax calls per subscriber

4.3 Trunk dimensioning

Spy functions and probes located in the different BSC and MSC allow the conversion of point-to-point traffic load into trunk requirements. In other word, at the end of a simulation it is possible to determine the number of trunks needed to handle a simulated traffic.

It is also possible to modified an existing network configuration depending on the traffic simulated by the MS-GEN. This is particularly interesting when simulating a blocking traffic. In such case, the network designer can use the facilities of the OPNET software in order to modify the existing topology by adding new trunk groups.

Moreover, the variation of trunks load can be viewed using the statistical an «***Analysis***» package of OPNET. In other words the following results can be obtained at the end of a simulation:

- Variation of the number of TCH allocated between the BSS and a MSC during the day
- Variation of the number of locup performed by the MS during the day
- Variation of the number of TCH allocated between a MSC and the PSTN
- Variation of the SDCCH load of each RELAY (BTS).

Finally, NETPLAN is able to present the graphical results of the trunk requirements of different network configuration scenarios.

4.4 Equipment dimensioning

For a given user population, it is possible to count the number of requests to the HLR as well as to the VLR and hence deduce the signalling transfer processing for these GSM entities.

NETPLAN takes into account the capacity of the equipment treating voice and signalling data. Moreover, it is also possible to modify the GSM network entities capacity in the Gsm.h file as follows:

- MSC_CAPACITY which is the maximum number of MS-MSC connections
- BSC_STORING_CAPACITY which is the maximum number of RXLEV/RXQUAL values stored for averaging
- GMSC_PROCESSING_CAPACITY which is the maximum number of incoming calls that can be processed by the GMSC.
- PSTN_PROCESSING_CAPACITY which is the maximum number of MT calls that the PSTN node can simulates.

4.5 Optimization

According to the results obtained, the network designer is able to state on an optimized network configuration.

However, the decision is not taken by NETPLAN; additional functions need to be added if NETPLAN should propose automatically an optimized network configuration on the basis of the given input parameters of the MS-GENs.

4.6 Simulation of a whole GSM network

As NETPLAN is an extension of the WINES project, a whole GSM network can be simulated at the protocol level as described in Section 2.

Therefore, NETPLAN can help the network designer to:

- Analyse the network reaction during an overloading situation
- Know the blocking percentage of any trunk group for a given traffic load

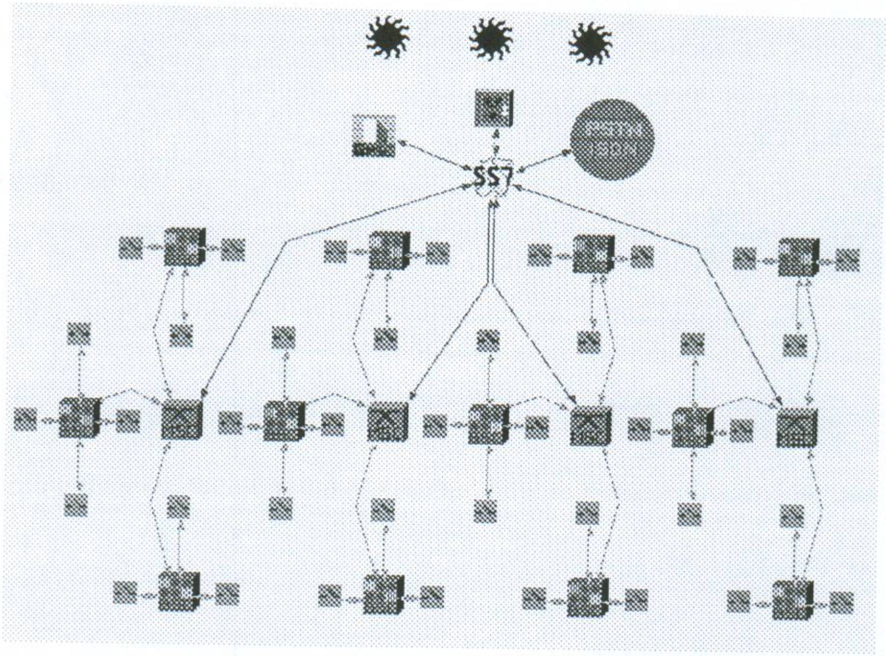

Figure 3 : Simulation of a GSM Network

4.7 Common channel signalling dimensioning

We are able to dimension each signalling link at each interface[1]. As all signalling protocols have been implemented, it is possible to determine the occupancy of each link and to notice the capacity limits for a given population of mobile users.

NETPLAN benefits from the graphical user interface facilities provided by OPNET and therefore can allow the network designer to specify the following parameters:

- Rate of each trunks (2 Mbits/s or 64 Kbits/s)
- Signalling capacity limits for each GSM equipment (see Gsm.h)
- Signalling link occupancy for each signalling link

However, it is not possible for the moment to have the following functionalities:

- Setting of specific rules for each equipment
- Definition of signalling restrictions of the network (i.e, whether or not an equipment has the right to communicate with another one)
- Cost-distance computations per signalling link

Note that NETPLAN is not able to recommend the optimal Signalling Transfer Point location from the given signalling network configuration.

1. For instance the data rate

4.8 Graphical Network Display

NETPLAN exploit the facilities of OPNET and thus can provide the following possibilities:

- Graphical Display of the network configurations. Figure 3 is an example of a network configuration
- Graphical display of each equipment. At the OPNET node level it is possible to show the different processors used as well as the trunk groups and the signalling links connecting the GSM entity with another equipment
- The OPNET editor allows the definition and the modification of GSM topologies and architectures
- Graphical display of the trunk dimensioning values
- Statistical display of data in the form of histograms or graphs representing the evolution of switch capacity, number of activated MS, traffic load of a specific equipment.

A example of a graphical display result is depicted in Figure 4.

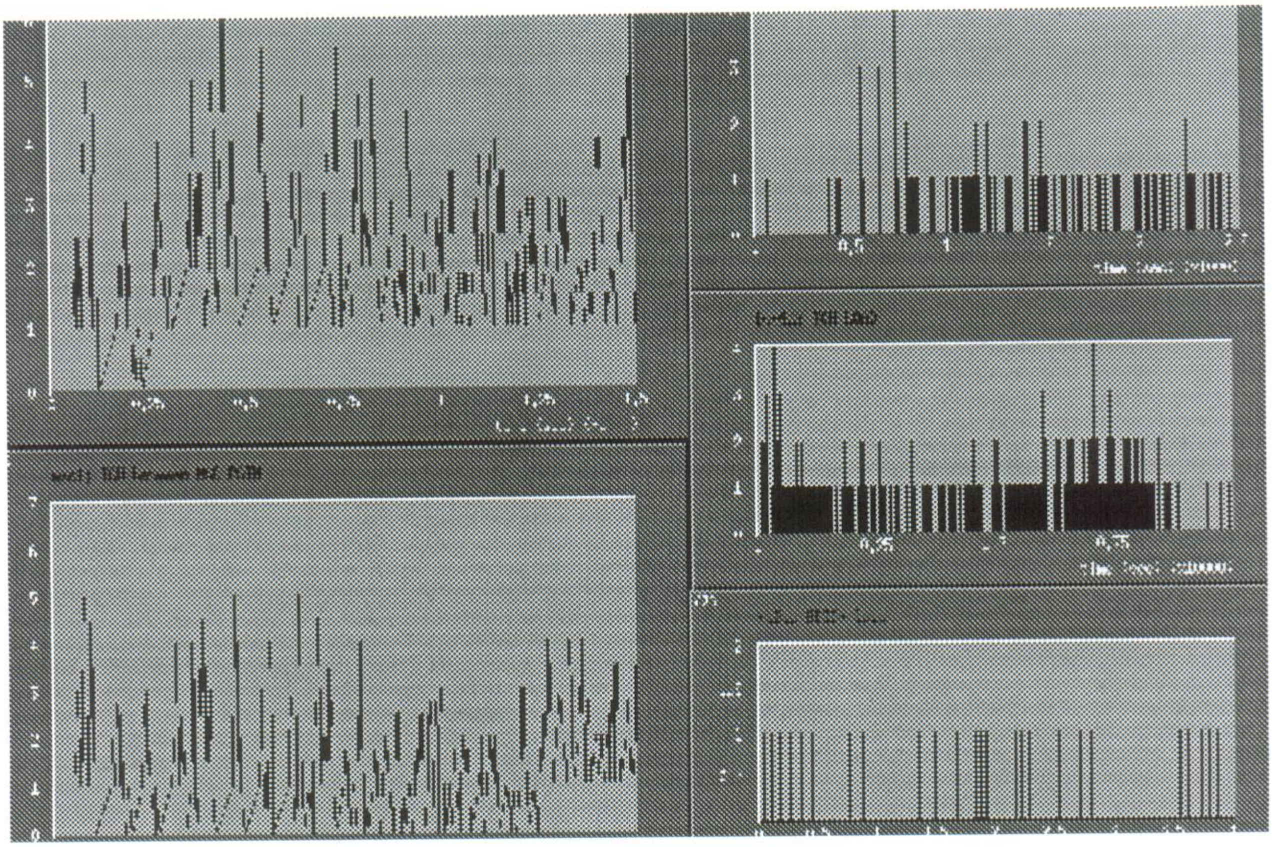

Figure 4 : Traffic Channels allocated by an MSC and a BSC

4.9 Reporting

NETPLAN uses OPNET «Analysis» tool to provide a report to the network designer. For the moment, the user cannot obtain a text report. However, it is possible to combine different results in a single graph such as the number of locups on the ordinate axis and the number of VLR request on the abscissa axis. The contents of the report is in fact a set of graphs copied from screen (postcript format) representing the following information:

- Traffic or signalling load in each interface of an equipment
- Total or partial load of an equipment

However, it is not possible for the moment to obtain a report in a text format containing trunk and signalling link requirement per equipment. This kind of results can be obtained by the network designer by analysing the obtained graphs (see Section 4.8). This analysis can be done by an external program which input can be a file containing graph data.

5 Conclusion

In conclusion, this report presents a new modular approach to the planning of the NSS and the BSS of mobile networks like GSM. NETPLAN includes WINES a network simulator which is able to simulate an entire GSM network as well as the mobile stations at the layer-3 protocol level. Various GSM network topologies (including the NSS and BSS) can be created and different components (mobile stations and mobile stations generators) are available for the generation of traffic at the MS level or the NSS level. The mobile station generators have been designed to be independent of the simulated network and can be used in other mobile network simulators.

We showed that NETPLAN can be of interest to network designers who can replay realistic scenarios based on observations made on a real network in order to find the best configuration for their GSM network.

Future work includes an enhancement of WINES in order to take into account supplementary services such as short messages service, voice mail and facsimile. The objectives will be to provide an accurate analysis on the impact of the introduction of a new service to an existing GSM network. Therefore, the missing functionalities in this version of NETPLAN will be addressed.

6 References

[1] D. Samfat and C. Bonnet «WINES: a Wireless Network Simulator for Network Management Applications», Technical Report No RR 96 023, Eurécom Institute, Sophia-Antipolis, France, May 1996

[2] D. Samfat and C. Bonnet «WINES: a Flexible GSM Simulator for Intrusion Detection and Network Planning Applications», Submitted to the Western Simulation Multiconference, Arizona, January 1997.

[3] D. Samfat, V. Devernay, C. Bonnet, «A GSM Simulation Platform for Intrusion Detection», Proceedings of ICC'95, Seatle, June 1995

[4] V. Devernay, «A GSM Simulation Platform for Intrusion detection», Internal Technical Description Report, Eurécom Institute, Sophia-Antipolis, France, May 1996.

[5] M. Mouly, M.B. Pautet, The GSM System for Mobile Communications, ISBN 2-9507190-0-7, 1993

[6] M. Baffulo ND P.J Kuhn, «Capacity and Performance Analysis for Signalling Networks Suporting UPT», Proceedings of the 8th ITC Specialist Seminar on Universal Personal telecommunication, Genova, Italy, Oct. 1992.

[7] J. Short, R. Bagrodia, L. Kleinrock, «Mobile Wireless Network Simulation», Proceedings of Mobicom'95, Berkley, CA, Nov. 1995.

[8] G. Willman, P.J Kuhn, «Performance Modelling of Signaling System No. 7», IEEE Communications Magazine, vol. 28, no 7, pp 44-56, July 1990.

[9] D. Dimitrijevic, J. Vucetic, «Design and Performance Analysis of the Algorithms for Channel Allocation in Cellular Networks», IEEE Transactions on Vehicular Technology, vol. 42, pp 526-534, 1993.

[10] Bagrodia, R. and W-L Liao, «Maisie: A Language for Design of Efficient Discrete-Event Simulations», IEEE Transactions on Software Engineering, Apr. 1994.

[11] GSM Recommendation 3.05, Signalling Requirements Relating to Routing of calls to Mobile Subscribers, ETSI Standard, Feb. 1992

[12] GSM Recommendation 3.09, Handover Procedures, ETSI Standard, Feb. 1992

[13] GSM Recommendation 3.12, Location Registration Procedures, ETSI Standard, Feb. 1992

[14] GSM Recommendation 4.08, Mobile Radio Interface Layer-3 Specification, ETSI Standard, Feb. 1992

[15] GSM Recommendation 9.02, MAP Specification, ETSI Standard, Feb. 1992

[16] GSM Recommendation 4.06, MS-BSS Data Link Layer Specification, ETSI Standard, Feb. 1992

[17] OPNET Network Simulation Software, MIL 3, Inc. 3400 International Drive, Washigton, DC 20008, Release 2.4.A